**一.Maven的作用**

在开发中，为了保证编译通过，我们会到处去寻找jar包，当编译通过了，运行的时候，有时也会报"ClassNotFoundException"。

每个Java项目的目录结构都没有一个统一的标准，配置文件到处都是，单元测试代码到底应该放在那里也没有一个权威的规范。

因此，我们就要用到Maven（使用Ant也可以，不过编写Ant的xml脚本比较麻烦）----一个**项目管理工具**。

Maven主要做了两件事：统一开发规范与工具 / 统一管理jar包

**Maven的目录结构：**

* main目录下是项目的主要代码，test目录下存放测试相关的代码
* 编译输出后的代码会放在target目录下
* src/main/java下存放Java代码，src/main/resources下存放配置文件
* 这里没有webapp，Web项目会有webapp目录，webapp下存放Web应用相关代码
* pom.xml是Maven项目的配置文件

**二．pom.xml**

基础的pom.xml是这样的：

<project xmlns**="http://maven.apache.org/POM/4.0.0"** xmlns:xsi**="http://www.w3.org/2001/XMLSchema-instance"** xsi:schemaLocation**="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4\_0\_0.xsd"**>  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>zhangys</groupId>  
 <artifactId>ce</artifactId>  
 <packaging>war</packaging>  
 <version>1.0-SNAPSHOT</version>  
 <name>ce Maven Webapp</name>  
 <url>http://maven.apache.org</url>  
 <dependencies>  
 </dependencies>  
 <build>  
 <finalName>ce</finalName>  
 </build>  
</project>

1. **modelVersion**

指定当前Maven模型的版本号

1. **groupId**

顾名思义，这个应该是公司名或是组织名。一般来说groupId是由三个部分组成，每个部分之间以"."分隔，第一部分是项目用途，比如用于商业的就是"com"，用于非营利性组织的就是"org"；第二部分是公司名，比如"tengxun"、"baidu"、"alibaba"；第三部分是你的项目名

**3、artifactId**

　　可以认为是Maven构建的项目名，比如你的项目中有子项目，就可以使用"项目名-子项目名"的命名方式

**4、version**

版本号，SNAPSHOT意为快照，说明该项目还在开发中，是不稳定的版本。在Maven中很重要的一点是，**groupId、artifactId、version三个元素生成了一个Maven项目的基本坐标**。

**5、packing**

　　项目打包的类型，可以使jar、war、rar、ear、pom，默认是jar

**6、dependencies和dependency**

　　前者包含后者。前面说了，Maven的一个重要作用就是统一管理jar包，为了一个项目可以build或运行，项目中不可避免的，会依赖很多其他的jar包，在Maven中，这些依赖就被称为dependency。

　　说到这里，就有一个**本地仓库**和**远程仓库**的概念了。官方下载暗转的Maven的本地仓库的配置在"%MAVEN\_HOME%\conf\settings.xml"里面，找一下"localRepository"就可以了；MyEclipse / IDEA默认的本地仓库的地址在"{user.home}/.m2/repository"路径下。

　　本地仓库和远程仓库是这样的，**Maven工程首先会从本地仓库中获取jar包，当无法获取指定jar包时，本地仓库会从远程仓库（中央仓库）中下载jar包，并放入本地仓库以备将来使用**。

　　举个例子，比方说我的项目中用到了MyBatis，那么可以这么配置：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <dependencies>      <dependency>          <groupId>org.mybatis</groupId>          <artifactId>mybatis</artifactId>          <version>3.2.5</version>      </dependency>  </dependencies> |

    之前有说过groupId、artifactId、version唯一标识一个Maven项目，有了这三个元素，我们就可以去远程仓库下载MyBatis3.2.5.jar到本地仓库了。

**7、properties**

　　properties是用来定义一些配置属性的，如project.build.sourceEncoding（项目构建源码编码方式），可以设置为UTF-8，防止中文乱码，也可定义相关构建版本号，便于日后统一升级。

**8、build**

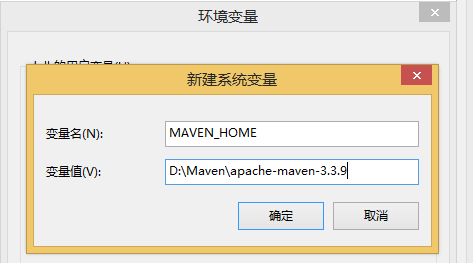
build表示与构建相关的配置，比如build下有finalName，表示的就是最终构建之后的名称。

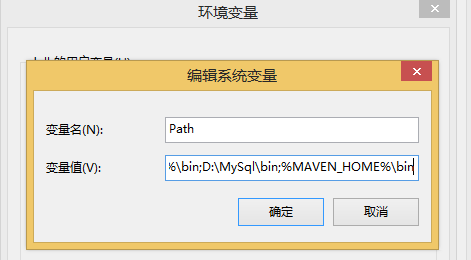
**三.Maven环境配置**

常用的开发工具Idea、MyEclipse里面都已经集成了Maven了，不过最好是从官网下一个配置到自己电脑里，开发工具里的可能有少许的Bug。

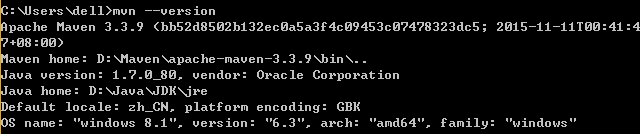
首先去Maven官网，下载Maven的包，地址为http://maven.apache.org/download.cgi

下载完解压，然后配置一下环境变量，和JDK的环境变量配置类似：





这样配置完就可以了，验证一下，windows+r打开命令窗口，输入"mvn --version"，如果有下面的内容这表示Maven配置OK：



**四.Maven常见问题**